SQL CASE STUDY:

Danny wants to use the data to answer a few simple questions about his customers, especially about their visiting patterns, how much money they've spent and also which menu items are their favourite. Having this deeper connection with his customers will help him deliver a better and more personalised experience for his loyal customers.

He plans on using these insights to help him decide whether he should expand the existing customer loyalty program - additionally he needs help to generate some basic datasets so his team can easily inspect the data without needing to use SQL.

Danny has shared with you 3 key datasets for this case study:

- `sales`
- `menu`
- `members`

Table 1: sales

The `sales` table captures all `customer_id` level purchases with an corresponding `order_date` and `product_id` information for when and what menu items were ordered.

customer_id	order_date	product_id
A	2021-01-01	1
A	2021-01-01	2
A	2021-01-07	2
A	2021-01-10	3
A	2021-01-11	3
A	2021-01-11	3
B	2021-01-01	2

B	2021-01-02	2
B	2021-01-04	1
B	2021-01-11	1
B	2021-01-16	3
B	2021-02-01	3
C	2021-01-01	3
C	2021-01-01	3
C	2021-01-07	3

Table 2: menu

The `menu` table maps the `product_id` to the actual `product_name` and `price` of each menu item.

product_id	product_name	price
1	sushi	10
2	curry	15
3	ramen	12

Table 3: members

The final `members` table captures the `join_date` when a `customer_id` joined the beta version of the Danny's Diner loyalty program.

customer_id	join_date
A	2021-01-07
B	2021-01-09

---1. What is the total amount each customer spent at the restaurant?

```
 select sales.customer_id as customers, sum(menu.price) as
total_amount
 from sales left join menu on sales.product_id=menu.product_id group by
sales.customer_id;
```

customers	total_amount
B	74
C	36
A	76

---2. How many days has each customer visited the restaurant?

```
select customer_id,count(customer_id) as days_visited from sales group by
customer_id order by customer_id;
```

customer_id	days_visited
A	6
B	6
C	3

---3. What was the first item from the menu purchased by each customer?

```
 select distinct sales.customer_id as
customers,first_value(menu.product_name) over(partition by
sales.customer_id order by sales.order_date) as first_product from sales
left join menu on
 sales.product_id=menu.product_id order by customers;
```

customers	first_product
A	curry
B	curry
C	ramen

4.What is the most purchased item on the menu and how many times was it purchased by all customers?

```
select  menu.product_name as product , count(*) as occurrence

 from sales left join menu on sales.product_id=menu.product_id
group by menu.product_name order by occurrence desc limit 1;
```

```
product | occurrence

---------+-----------

 ramen   |        8
```

5.Which item was the most popular for each customer?

```
WITH CustomerProductPurchases AS (

SELECT s.customer_id, s.product_id, m.product_name, COUNT(*) AS
total_purchases FROM sales s JOIN menu m ON s.product_id =
m.product_id GROUP BY s.customer_id, s.product_id,
m.product_name ),

 RankedProducts AS (

SELECT customer_id, product_id, product_name, total_purchases,
RANK() OVER (PARTITION BY customer_id ORDER BY
total_purchases DESC) AS rank FROM CustomerProductPurchases )

SELECT customer_id, product_id, product_name, total_purchases
FROM RankedProducts WHERE rank = 1;
```

```
customer_id | product_id | product_name | total_purchases

------------+------------+--------------+----------------

 A          |          3 | ramen        |              3

 B          |          1 | sushi        |              2

 B          |          3 | ramen        |              2

 B          |          2 | curry        |              2
```

C | 3 | ramen | 3

6.Which item was purchased first by the customer after they became a member?

```
WITH membership AS (

SELECT s.customer_id AS customer, s.order_date AS first_order,
s.product_id AS id, m.product_name AS name, mem.join_date AS
joining

 FROM sales s LEFT JOIN menu m ON s.product_id = m.product_id
LEFT JOIN members mem ON s.customer_id = mem.customer_id ),

ranked_membership AS (

SELECT customer, first_order, name, joining, RANK() OVER
(PARTITION BY customer ORDER BY first_order) AS rank_ FROM
membership WHERE first_order >= joining AND joining IS NOT NULL )

SELECT customer, first_order, name FROM ranked_membership
WHERE rank_ = 1;
```

customer | first_order | name

A | 2021-01-07 | curry

B | 2021-01-11 | sushi

7.Which item was purchased just before the customer became a member?

```
WITH PreMembershipPurchases AS (

 SELECT s.customer_id, s.order_date, s.product_id, m.product_name,
mem.join_date FROM sales s JOIN members mem ON s.customer_id
= mem.customer_id JOIN menu m ON s.product_id = m.product_id
WHERE s.order_date < mem.join_date ),
```

RankedPurchases AS (

SELECT customer_id, order_date, product_id, product_name, RANK() OVER (PARTITION BY customer_id ORDER BY order_date DESC) AS rn FROM PreMembershipPurchases)

SELECT customer_id, product_id, product_name, order_date FROM RankedPurchases WHERE rn = 1;

8.What is the total items and amount spent for each member before they became a member?

WITH PreMembershipPurchases AS (

SELECT s.customer_id, s.product_id, COUNT(*) AS total_items, SUM(m.price) AS total_amount

FROM sales s JOIN menu m ON s.product_id = m.product_id

JOIN members mem ON s.customer_id = mem.customer_id

WHERE s.order_date < mem.join_date GROUP BY s.customer_id, s.product_id)

SELECT p.customer_id, COUNT(*) AS total_items, SUM(total_amount) AS total_amount_spent FROM PreMembershipPurchases p GROUP BY p.customer_id;

customer_id | total_items | total_amount_spent

----------+----------+------------------

A | 2 | 25

B | 2 | 40